

## Program szkolenia:

# Programowanie równoległe w .NET Framework

## Informacje:

<b>Nazwa:</b>	<b>Programowanie równoległe w .NET Framework</b>
<b>Kod:</b>	<b>NETplatform-.Net Parallel</b>
<b>Kategoria:</b>	Platforma .NET
<b>Grupa docelowa:</b>	architekci developerzy
<b>Czas trwania:</b>	3 dni
<b>Forma:</b>	50% wykłady / 50% warsztaty

Szkolenie ma za zadanie zaznajomić słuchaczy z możliwościami jakie niesie nowa wersja Microsoft .NET Framework 4.0 w dziedzinie obliczeń równoległych.

W trakcie kursu słuchacze poznają podstawowe założenia, zasady, narzędzia i dobre praktyki programowania równoległego na platformie .NET przy wykorzystaniu języka C#. Podczas praktycznej części szkolenia uczestnicy będą mieli okazję tworzyć, uruchamiać, testować oraz debugować kod równoległy.

Obecnie praktycznie każdy komputer, laptop, a nawet telefon komórkowy jest wyposażony w procesor zawierający kilka niezależnie pracujących od siebie rdzeni. Wykorzystanie zalet tych procesorów wykorzystując klasyczne metody programowania (współdzielona pamięć, synchronizacja, blokowanie, wiele wątków) jest bardzo trudne i nastęcza wiele problemów.

W celu ułatwienia tworzenia kodu równoległego, który może być wykonywany niezależnie przez kilka rdzeni, w .NET Framework wprowadzono tzw. Parallel Extensions. Praktycznie jest to całkowicie nowy paradygmat tworzenia oprogramowania, które może zostać zrównoleglone na dowolną ilość rdzeni /procesorów przy relatywnie niewielkim nakładzie pracy oraz czytelniejszym i prostszym w utrzymaniu kodzie.

### Wymagania

Uczestnicy znają podstawy programowania w języku C# oraz potrafią obsługiwać środowisko programistyczne Visual Studio 2010.

Na stacjach roboczych jest zainstalowane środowisko Visual Studio 2010. Stacje robocze są wyposażone w procesory co najmniej dwurdzeniowe.

## Zalety szkolenia:

- Wykorzystanie potencjału sprzętu
- Praktyczne przykłady

## Szczegółowy program:

### 1. Wprowadzenie do Parallel Extensions

1.1. Geneza powstania

1.2. Korzyści

### 2. Task Parallelism - Task Parallel Library

2.1. Klasyczne podejście imperatywne

2.2. TPL i kod równoległy

### 3. Data Parallelism – PLINQ

3.1. Wprowadzenie do LINQ

3.2. Podejście imperatywne

3.3. Podejście deklaratywne - Data Parallelism i PLINQ

### 4. Coordination Data Structures

4.1. Współbieżne i skalowalne struktury danych

4.2. Metody synchronizacji

4.3. Leniwa inicjalizacja obiektów

### 5. Parallel Diagnostics Tools

5.1. Debugging

5.2. Profiling

### 6. Custom Partitioners for PLINQ and TPL

### 7. Task Factories

### 8. Task Schedulers

### 9. Lambda Expressions in PLINQ and TPL

### 10. Zagadnienia zaawansowane

10.1. Istotne aspekty systemu

10.1.1. Działanie systemu operacyjnego

10.1.2. Działanie procesora

10.1.3. Hierarchia pamięci

10.2. Specyficzne zachowania kodu wykonywanego równoległe

10.3. Synchronizacja natywna - prymitywy .net

10.4. Koncepcje lock-free memory model

10.5. Problemy, wzorce i ich rozwiązania

10.6. Reactive Extensions - przetwarzanie strumienia asynchronicznych zdarzeń

10.7. async i await.